



0 8 7 7 6 3 8 6 8

PCT/FR 00 / 0 1 8 1 4

B R E V E T D ' I N V E N T I O N

REC'D 02 AUG 2000

WIPO PCT

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

4

COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 05 JUIL 2000

REGLE 17.1.a) OU b)

DOCUMENT DE PRIORITÉ

PRÉSENTÉ OU TRANSMIS
CONFORMÉMENT À LA
RÈGLE 17.1.a) OU b)

Pour le Directeur général de l'Institut
national de la propriété industrielle
Le Chef du Département des brevets

Martine PLANCHE

BEST AVAILABLE COPY

INSTITUT
NATIONAL DE
LA PROPRIÉTÉ
INDUSTRIELLE

SIEGE

26 bis, rue de Saint Petersburg
75800 PARIS Cédex 08
Téléphone : 01 53 04 53 04
Télécopie : 01 42 93 59 30



This Page Blank (uspto)



26 bis, rue de Saint Pétersbourg
75200 Paris Cedex 08
Téléphone : 01 53 04 53 04 Télécopie : 01 42 93 59 30

REQUÊTE EN DÉLIVRANCE

Confirmation d'un dépôt par télécopie

BEST AVAILABLE COPY

<p>Reserve a l'INPI</p> <p>DATE DE REMISE DES PIÈCES 30 JUIN 1999</p> <p>N° D'ENREGISTREMENT NATIONAL 9908409</p> <p>DÉPARTEMENT DE DÉPÔT 75 INPI PARIS 6</p> <p>DATE DE DÉPÔT 30 JUIN 1999</p>		<p>1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE</p> <p>BULL S.A. Monsieur Bernard CORLU / PC 58F35 68, route de Versailles 78434 LOUVECIENNES CEDEX</p> <p>n° du pouvoir permanent PG 4280 références du correspondant FR3830/BC téléphone 01 39.66.61.76</p>					
<p>2 DEMANDE Nature du titre de propriété industrielle</p> <p><input checked="" type="checkbox"/> brevet d'invention <input type="checkbox"/> demande divisionnaire</p> <p><input type="checkbox"/> certificat d'utilité <input type="checkbox"/> transformation d'une demande de brevet européen</p> <p><input type="checkbox"/> demande initiale</p> <p><input type="checkbox"/> brevet d'invention</p> <p>Établissement du rapport de recherche <input type="checkbox"/> différé <input checked="" type="checkbox"/> immédiat</p> <p>Le demandeur, personne physique, requiert le paiement échelonné de la redevance <input type="checkbox"/> oui <input checked="" type="checkbox"/> non</p> <p>Titre de l'invention (200 caractères maximum)</p> <p>Procédé de sécurisation du traitement d'une information sensible dans un module de sécurité monolithique, et module de sécurité associé.</p>							
<p>3 DEMANDEUR (S) n° SIREN 3.2.9.5.5.6.1.4.6 code APE-NAF B.3.2.1</p> <p>Nom et prénoms (souligner le nom patronymique) ou dénomination</p> <p>BULL CP8</p>		<p>Forme juridique</p> <p>S.A.</p>					
<p>Nationalité (s) Française</p> <p>Adresse (s) complète (s)</p> <p>BULL CP8 BP 45 68, route de Versailles 78430 LOUVECIENNES</p>		<p>Pays</p> <p>FRANCE</p>					
<p>4 INVENTEUR (S) Les inventeurs sont les demandeurs <input type="checkbox"/> oui <input checked="" type="checkbox"/> non En cas d'insuffisance de place, poursuivre sur papier libre <input type="checkbox"/> Si la réponse est non, fournir une désignation séparée</p>							
<p>5 RÉDUCTION DU TAUX DES REDEVANCES <input type="checkbox"/> requise pour la 1ère fois <input type="checkbox"/> requise antérieurement au dépôt : joindre copie de la décision d'admission</p>							
<p>6 DÉCLARATION DE PRIORITÉ OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE</p> <table border="1"><thead><tr><th>pays d'origine</th><th>numéro</th><th>date de dépôt</th><th>nature de la demande</th></tr></thead><tbody></tbody></table>				pays d'origine	numéro	date de dépôt	nature de la demande
pays d'origine	numéro	date de dépôt	nature de la demande				
<p>7 DIVISIONS antérieures à la présente demande n° date n° date</p>							
<p>8 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE (nom et qualité du signataire)</p> <p>Bernard CORLU Mandataire - </p>		<p>SIGNATURE DU PRÉPOSÉ À LA RÉCEPTION SIGNATURE APRÈS ENREGISTREMENT DE LA DEMANDE À L'INPI</p>					

170-17 ou 9 janvier 1976 relative à l'informatique aux fins de rectification pour les données vous concernant auprès de l'INPI. Elle garantit un droit d'accès et de rectification pour les données vous concernant auprès de l'INPI.

DA 440 3 78/100



BREVET D'INVENTION CERTIFICAT D'UTILITE

DÉSIGNATION DE L'INVENTEUR

(si le demandeur n'est pas l'inventeur ou l'unique inventeur)

N° D'ENREGISTREMENT NATIONAL

 7908409

DEPARTEMENT DES BREVETS

26bis, rue de Saint-Petersbourg

75800 Paris Cédex 08

Tél. : 01 53 04 53 04 - Télécopie : 01 42 93 59 30

FR 3830/BC

TITRE DE L'INVENTION :

Procédé de sécurisation du traitement d'une information sensible dans un module de sécurité monolithique, et module de sécurité associé.

LE(S) SOUSSIGNÉ(S)

BULL S.A.

DÉSIGNE(NT) EN TANT QU'INVENTEUR(S) (indiquer nom, prénoms, adresse et souligner le nom patronymique) :

Hazard Michel
27 rue des Harias
78124 MAREIL SUR MAULDRE
France

NOTA : A titre exceptionnel, le nom de l'inventeur peut être suivi de celui de la société à laquelle il appartient (société d'appartenance) lorsque celle-ci est différente de la société déposante ou titulaire.

Date et signature (s) du (des) demandeur (s) ou du mandataire

Louveciennes, le 25 juin 1990

C rlu Bernard (mandataire)

Procédé de sécurisation du traitement d'une information sensible dans un module de sécurité monolithique, et module de sécurité associé

La présente invention concerne un procédé et un dispositif pour
5 accroître la sécurité d'un module de sécurité monolithique comprenant un microprocesseur et agencé pour exécuter un programme à sécuriser. Un programme à sécuriser est un ensemble d'instructions dont l'exécution doit répondre à des critères tels que : authentification de l'utilisateur, confidentialité des données échangées, authenticité d'une transaction et sa validité, de façon
10 générale, le traitement de données d'applications dans lesquelles des droits et obligations d'un usager sont contrôlés. La présente invention vise à doter le module de sécurité de moyens de détection et de parade contre des interventions extérieures frauduleuses pour accéder à des données sensibles, par contournement de contrôles préalables connus en eux-mêmes.

15 Le terme « module de sécurité » monolithique recouvre tout circuit électronique constitué d'une seule puce électronique et enrobé dans une enceinte, le circuit comprenant une unité de traitement, au moins une mémoire et des périphériques tels que des circuits d'entrée/sortie, contrôleur d'interruptions, etc.... Le module de sécurité peut prendre l'aspect d'un circuit
20 intégré ou d'un objet portatif du type carte à puce utilisée par exemple, dans le domaine bancaire, dans les radiotéléphones mobiles, les décodeurs de télévision à péage, la santé, les transports.

Les modules de sécurité selon l'invention comportent au moins un microprocesseur, une mémoire contenant un programme et des moyens
25 d'entrée/sortie pour communiquer avec l'extérieur. La mémoire contient au moins un programme d'application qui peut être inscrit dans une mémoire de type ROM lors de la fabrication du circuit, ou écrit par la suite dans une mémoire programmable. De façon générale, le programme contient les instructions exécutées par le microprocesseur, le transfert des instructions de
30 la mémoire morte ou non volatile au microprocesseur s'effectuant par un bus

de données couplé à un bus d'adresses. Si les chemins de bus sont trop longs, le constructeur du circuit dispose des circuits amplificateurs aux extrémités des bus afin que le signal binaire puisse se propager correctement sur toute sa longueur..

5

Les circuits amplificateurs exigent une forte impédance d'entrée ce qui les rend sensibles aux perturbations extérieures. Soumis à un rayonnement de particules élémentaires, ils peuvent se saturer pendant un certain temps et, quelle que soit la donnée en entrée, n'émettre sur le bus que 0 volt ou +Vcc, c'est-à-dire un "0" ou un "1" binaire. Un tel rayonnement peut être constitué de rayons alpha, X, d'ions chargés positivement ou négativement selon que l'on veut une sortie de l'amplificateur à "0" ou "1".

Un fraudeur en possession d'une carte authentique peut essayer de déjouer les dispositifs de sécurité afin de profiter de services de façon abusive : il va donc soumettre le module de sécurité à de tels rayonnements en espérant perturber son fonctionnement à son avantage. La source d'émission peut être suffisamment courte pour ne perturber l'exécution que d'une ou plusieurs instructions. Cette perturbation peut substituer à la valeur lue dans la mémoire du programme, une autre valeur imposée par l'émission du rayonnement. Ainsi, le déroulement du programme est différent et le fraudeur peut en tirer avantage.

Une première parade à ce type d'attaque est d'installer des capteurs de rayonnement, dès que ceux-ci détectent un rayonnement anormal, ils positionnent un drapeau qui peut être lu par le programme. Une telle solution est décrite dans le brevet US 5.465.349, équivalent américain du brevet FR 2 668 274. Mais les fraudeurs ont amélioré les moyens d'investigation et utilisent des émetteurs de rayonnement extrêmement fins. En pointant le rayonnement uniquement sur les amplificateurs ou un groupe d'amplificateurs, les capteurs ne détectent plus la fraude. En tout état de cause, cette parade est incapable

de détecter une intervention à l'aide de micro-pointes et, de toute façon, on peut perturber la séquence de lecture des capteurs pour que le programme s'exécute comme si rien d'anormal n'était détecté.

5 L'invention vise à détecter des altérations dans la transmission des instructions et des données entre une mémoire et un microprocesseur. Le problème que vise à résoudre l'invention est la détection d'une intervention extérieure visant à perturber le déroulement d'un programme en modifiant la valeur des signaux échangés sur les bus d'un micro-contrôleur.

10

Le problème est résolu selon l'invention en prévoyant des mesures permettant aux modules de sécurité de vérifier si les informations transitent correctement au sein du module et si les programmes ont été exécutés intégralement. Dans la négative, l'exécution normale du programme est interrompue. De façon optionnelle, toute utilisation ultérieure du module est interdite car le module s'est rendu compte qu'il est utilisé de façon illicite.

15

Plus précisément, l'invention revendique un procédé de sécurisation du traitement d'une information sensible dans un module de sécurité de structure monolithique, le module comportant des moyens de traitement de l'information et des moyens de mémorisation d'informations susceptibles d'être traitées par lesdits moyens de traitement, caractérisé en ce qu'il comprend les étapes suivantes :

20

- sélectionner une information sensible dans les moyens de
- 25 mémorisation ;
- déterminer une condition particulière d'intégrité de la dite information ;
- lire l'information et la transmettre aux moyens de traitement ;
- contrôler lors du traitement de l'information que la condition particulière est satisfaite ;

- bloquer le traitement de l'information au cas où la condition particulière n'est pas satisfaite.

D'autres caractéristiques et avantages de l'invention apparaîtront au cours de la description suivante de quelques modes de réalisation préférés mais non limitatifs, en regard des dessins annexés, sur lesquels :

La figure 1 représente le schéma d'un système classique à microprocesseur.

La figure 2 montre le jeu d'instructions du microprocesseur MOTOROLA 6805.

La figure 3 montre le jeu modifié d'instructions du microprocesseur selon l'invention.

La figure 4 montre le schéma synoptique du circuit modifié selon l'invention.

La figure 5 montre une partie du schéma synoptique du circuit modifié selon une amélioration de l'invention.

La figure 6 montre une partie de la mémoire de programme dotée d'étapes de mise à jour et de modification d'un indicateur permettant de détecter des perturbations extérieures.

La figure 7 montre une partie de la mémoire de programme selon une variante de l'invention.

Le schéma électronique d'un système à microprocesseur est représenté à la figure 1. Ce schéma est extrait du livre « Architecture de l'Ordinateur » de Andrew TANENBAUM, publié par InterEditions. Le système comprend de façon connue en soi les éléments suivants : un microprocesseur 31, une mémoire RAM 32, une mémoire de programme EPROM 33 qui contient le programme exécutable et des moyens d'entrée/sortie 34. Les connexions de ces divers éléments s'effectuent par deux bus, le bus d'adresses 35 et le bus de données 36. On a coutume de désigner par « bus de contrôle » le bus qui véhicule

l'ensemble des signaux tels que les signaux d'horloge, les signaux de lecture et d'écriture,....

Selon l'opération exécutée, ces éléments peuvent être en mode
5 réception d'informations ou en mode émission d'informations. Lorsque le microprocesseur lit une instruction dans un octet à une certaine adresse, les huit cellules de la mémoire ROM constituant l'octet correspondant à cette adresse, émettent la valeur inscrite dans les cellules à travers le bus de données qui la transmet au microprocesseur, ce dernier étant en mode
10 réception. Inversement, lorsque le microprocesseur émet une donnée vers la mémoire, le bus de données est alors en mode émission. Le fonctionnement interne du composant est détaillé dans le brevet US 4.382.279 équivalent américain du brevet FR 2 461 301 .

15 Chaque microprocesseur possède un jeu spécifique d'instructions. Le microprocesseur MOTOROLA 6805 traite des données de huit bits en interne. Son bus de données comprend 8 lignes et le code opération de ses instructions est codé sur 8 bits. . A titre d'exemple, le jeu d'instruction du microprocesseur M6805 et M1468805 extrait de la documentation
20 « Microcomputer/microprocessor User's manual » est représenté à la figure 2. Une instruction pour ce type de microprocesseur est codée sur 8 bits, donc il existe 256 codes différents mais ceux-ci ne sont pas tous exécutables par le microprocesseur. Certains codes ne sont pas implémentés et correspondent à des trous dans le tableau de la figure 2, par exemple les codes : 90H, 31H,
25 41H, 51H, 61H, 71H, 91H, 23H, 24H, 25H, 26H, 27H, 28H, 29H, 93H, 94H, etc... Si le microprocesseur lit un code opération non implémenté, son exécution n'est pas garantie. Ainsi, le microprocesseur peut passer à l'instruction suivante sans rien faire d'autre.

Comme on le constate sur la figure 2, le code "00" du jeu d'instructions du microprocesseur MOTOROLA 6805 correspond à l'instruction « BRSET0 » et le code "FF", à l'instruction « STX » en indexé. L'instruction « BRSET0 » occupe trois octets dans la mémoire de programme et s'exécute en dix cycles d'horloge, celui de l'instruction « STX » occupe un octet et s'exécute en cinq cycles d'horloge. Le microprocesseur possède un compteur ordinal contenant l'adresse de l'instruction dans la mémoire de programme qui est en cours d'exécution. Comme il a été dit en préambule, si un fraudeur envoie un rayonnement sur le bus, l'une ou l'autre de ces instructions sera exécutée à la place de celle effectivement lue dans la mémoire de programme. Dans le cas de l'instruction « BRSET0 », après son exécution, le compteur ordinal du microprocesseur est augmenté de trois unités et pointe donc trois octets plus loin dans la mémoire de programme. Pour l'instruction « STX », le compteur ordinal est augmenté d'une unité. En émettant le rayonnement pendant les dix coups d'horloge nécessaires à l'exécution du BRSET0, le microprocesseur va lire trois octets à «00», et traduire cela par « test si le bit 0 de l'octet 00 est à 1 et saut si c'est le cas ». A cause du rayonnement, la valeur 00 de l'octet à l'adresse 00 est lue, donc le saut ne va pas s'effectuer et le programme continue à l'instruction suivante. Ainsi, en soumettant le circuit à un rayonnement pendant un certain nombre de tops d'horloge, un fraudeur peut faire avancer artificiellement le compteur ordinal par sauts successifs de trois octets et empêcher l'exécution d'une séquence inscrite en mémoire de programme. Pour un microprocesseur de la famille 8051, fabriqué par la société INTEL et de nombreuses autres sociétés, le code opération dont la valeur binaire est 00 correspond à l'instruction NOP qui signifie « no operation ». Cette instruction n'utilise qu'un seul octet. Dans ce cas, le fraudeur peut faire avancer artificiellement le compteur ordinal octet par octet.

Une première solution consiste à interrompre le fonctionnement normal du microprocesseur lors d'une lecture d'instruction dont le code est « 00 » ou

« FF ». Le module devient muet, seule une mise hors tension suivie d'une mise sous tension du module peut faire repartir le microprocesseur. La figure 3 montre le jeu modifié d'instruction selon l'invention. La nouvelle instruction correspondant à « 00 » ou « FF » est appelée « FRAUDE » ou FRD en abrégé.

5 Par rapport au jeu d'instruction normal décrit dans sur la figure 2, des colonnes ont été inversées de telle sorte que les codes « 00 » et « FF » ne correspondent plus à des instructions exécutables. La colonne numéro « 0 » qui contenait l'instruction « BRSET0 » est transférée à la colonne 2, la colonne numéro « F » qui contenait l'instruction « STX » est transférée à la colonne A.

10 Le transfert des colonnes est fait en modifiant le masque du circuit, dans l'élément qui décode les quatre bits de poids forts du code opération des instructions. On aurait aussi pu modifier le décodage des quatre bits de poids faible du code opération, mais ce n'était pas nécessaire pour ce modèle de microprocesseur.

15 Le programme ne contient pas d'instruction « FRAUDE ». Le programmeur qui écrit un programme d'application, prend soin de ne pas implémenter de telles instructions dans son programme.

Comme le montre la figure 3, la solution peut être généralisée à tous les

20 codes opérations qui ne correspondent pas à une instruction reconnaissable par le microprocesseur. Ainsi, la nouvelle instruction « FRAUDE » est affectée à tous les codes non utilisés tels que : 01H, 02H, 03H, 04H, 05H, 06H, 0EH, 12H, 14H, 15H, 16H, 17H, 18H, 19H, 1AH, ...etc jusqu'à FFH, en tout 47 codes opération différents qui exécutent tous la même instruction « FRAUDE ».

25

Selon une variante, cette interruption déclenche une instruction microprogrammée dans le microprocesseur. Ce type d'instruction exécute une opération non réversible du type écriture d'un code en mémoire non volatile. Lors d'une prochaine mise sous tension, le circuit de gestion de la remise à

30 zéro teste la valeur de ce code en mémoire non volatile et bloque le

fonctionnement du microprocesseur. L'instruction microprogrammée déclenche un opération de blocage définitif du circuit. Les instructions microprogrammées ont l'avantage d'être résidentes dans le coeur du microprocesseur et donc, leur exécution ne peut être ni interrompue, ni altérée
 5 par un rayonnement agissant sur le bus. Il n'est donc pas possible de détecter l'exécution d'une instruction microprogrammée de blocage.

Une seconde solution permettant de détecter la perturbation d'une donnée sur un bus est d'implémenter un contrôle d'intégrité d'un bout à l'autre
 10 des bus. La figure 4 montre le schéma électronique avec son contrôleur d'intégrité de bus.

On rajoute aux huit lignes du bus de données 1 une neuvième ligne, notée PARITE 2 dont l'état logique correspond à la valeur de la parité calculée
 15 à partir des huit informations binaires présentes sur le bus de données. On a dit précédemment que les codes opération des diverses instructions d'un programme sont lus à partir de la mémoire ROM 3 ou de la mémoire programmable non volatile 4, EEPROM par exemple. Le signal de sortie d'une cellule de mémoire, dont la valeur représente la donnée binaire mémorisée par
 20 cette cellule, est trop faible pour atteindre le microprocesseur via le bus de donnée. Pour que ces signaux parviennent au microprocesseur, des circuits amplificateurs 5 et 6 sont installés juste après les cellules des mémoires EEPROM et ROM et avant le bus. Ces deux circuits ont une entrée de contrôle E qui permet d'activer leur sortie. Si le signal E a un niveau « 1 », les huit
 25 sorties ne sont pas actives. Elles sont dans un état haute impédance. La mémoire EEPROM étant accessible en écriture et en lecture, le circuit amplificateur 5 est bidirectionnel. Le sens de transfert des données est contrôlé par le microprocesseur par la ligne de contrôle READ/WRITE 18).

Les moyens de contrôle de l'intégrité de la donnée lue des cellules
 30 mémoires comprennent des générateurs de parité 7, 8 et 11, un comparateur

12 et une entrée d'interruption non masquable appelée « NMI » reliée au microprocesseur. Les générateurs de parité 7, 8 et 11 calculent la valeur de la parité de la donnée présente sur huit entrées. A titre d'exemple, le circuit 74HC280 fabriqué par National Semiconductor est un générateur de parité disponible sous la forme d'un circuit intégré. Sa structure est parfaitement intégrable au sein du circuit monolithique. Les générateurs de parité possèdent également une ligne de sortie Q dont l'état représente la valeur de la parité de la donnée appliquée sur les huit entrées et une ligne de contrôle E qui active la sortie Q lorsqu'on lui applique 0 volt. Dans ce cas, la ligne de sortie Q est à 0 volt si le nombre d'entrée à +Vcc est pair, soit à +Vcc si le nombre d'entrée à +Vcc est impair. Lorsque l'on applique +Vcc à l'entrée E), la ligne de sortie Q est dans un état haute impédance. Le générateur de parité 7 calcule la parité de la donnée sélectionnée dans la mémoire EEPROM, le générateur de parité 8 calcule la parité de la donnée sélectionnée dans la mémoire ROM et le générateur de parité 11 calcule celle présente dans le microprocesseur.

Le microprocesseur 9 possède aussi un circuit amplificateur 10 pour assurer la compatibilité des signaux transitant sur les bus de données, d'adresse et de contrôle. Ce circuit amplificateur relié au bus de données 10 est bidirectionnel, d'un côté, il amplifie les signaux de données que le microprocesseur envoie aux mémoires et aux périphériques, de l'autre côté il amplifie les signaux reçus par ces mêmes mémoires et périphériques afin que ces signaux soient traités correctement par les circuits internes du microprocesseur. La plupart des microprocesseurs ont un dispositif de pré-lecture « fetch » en anglais) qui permet dans le même cycle d'horloge d'exécuter une instruction et de lire le code opération de l'instruction suivante.

Le microprocesseur 9 contrôle le bus d'adresses, la valeur appliquée sur ce bus détermine quel type de périphérique est sélectionné. Pour diminuer le nombre de lignes de sélection, un décodeur d'adresse 13 reçoit en entrée le bus d'adresses et sélectionne les différents périphériques et mémoires par des

lignes spécialisées : la ligne appelée « EEPROM » sélectionne la mémoire EEPROM et, la ligne « ROM », la mémoire ROM. Chacune de ces deux lignes est reliée à la mémoire et au générateur de parité correspondant.

Les lignes EEPROM et ROM sont actives à l'état 0. La sortie de la porte ET 14 génère un signal de sélection commune des deux mémoires, si une des deux lignes EEPROM ou ROM est à « 0 », la ligne 15 est aussi à « 0 ». La ligne 15 est reliée à l'entrée E du générateur de parité 11 qui, de ce fait, est actif en même temps que l'un ou l'autre des générateurs de parité 7 ou 8. Si aucune mémoire n'est sélectionnée, les lignes 2 et 16 de sortie des générateurs sont à l'état haute impédance. Les lignes 2 et 16 de sortie des générateurs sont reliées respectivement à chacune des deux entrées du comparateur 12. La sortie Q du comparateur est reliée à une entrée d'interruption du microprocesseur (notée « NMI » sur la figure 4).

En utilisation, le microprocesseur lit un code opération représentatif d'une instruction dans la mémoire ROM. Il applique l'adresse du code sur le bus d'adresses ce qui rend actifs la ligne ROM et par voie de conséquence, les amplificateurs 6 et les générateurs de parité 8 et 11. Les amplificateurs envoient la donnée lue dans la mémoire ROM sur le bus de données, et le générateur de parité calcule la valeur de la parité de ladite donnée puis, envoie la valeur au comparateur 12 par la ligne PARITE 2. Le microprocesseur lit la donnée à travers les amplificateurs bidirectionnels 10. Le générateur de parité 11, activé par la ligne 15, calcule la valeur de la parité de la donnée lue par le microprocesseur et, l'envoie au comparateur par la ligne 16. Le comparateur 12 compare les deux valeurs : si elles sont égales, la sortie est au niveau « 1 », en cas de différence, la sortie est à « 0 ». Dans ce dernier cas, la donnée a subi une altération due très probablement à l'émission frauduleuse d'un rayonnement. La sortie du comparateur est reliée par la ligne 17 à l'entrée « NMI » du microprocesseur. Un niveau « 0 » déclenche une interruption non masquable qui provoque un déroutement du programme en cours vers une

routine d'interruption. L'activation du comparateur peut être effectuée lors de la phase de pré-lecture ; ainsi, la génération d'interruption est parfaitement synchronisée avec l'horloge de séquençement du microprocesseur.

L'opération est identique lorsque le microprocesseur effectue une
5 lecture de la mémoire EEPROM.

Dans un mode simplifié de l'invention, l'entrée « NMI » peut être assimilée à l'entrée RESET du microprocesseur. Lorsque l'entrée redevient inactive, le microprocesseur est débloqué et commence son programme de la même façon qu'une remise sous tension. Ainsi, une intervention extérieure ne
10 peut contrôler le microprocesseur qui, tant qu'il est soumis à ce rayonnement, est bloqué.

Par rapport à la première solution qui vise à vérifier qu'un code opération est correctement lu, cette seconde solution offre l'avantage de contrôler tout type de donnée : des codes opérations ou des données. Des
15 générateurs de parité peuvent également être installés sur le bus d'adresses de la même façon que pour le bus de données. Cette seconde solution offre donc aussi l'avantage de contrôler les signaux du bus d'adresses.

Dans une variante, un niveau « 0 » sur l'entrée « NMI » provoque
20 l'exécution d'un microprogramme. Pour éviter les phénomènes transitoires, la reconnaissance du niveau du signal sur l'entrée « NMI » s'effectue lors d'une transition de l'horloge du microprocesseur, transition au cours de laquelle les données analysées par le comparateur de signaux de parité sont valides. Un programme classique de gestion d'interruption écrit dans la mémoire ROM
25 aurait son exécution perturbée par le fraudeur. Par contre, selon l'invention, un microprogramme est résident dans le microprocesseur ; il ne fait pas appel à la lecture de données sur le bus ; son exécution ne peut donc être perturbée par un rayonnement agissant sur le bus. Le microprogramme, déclenché par l'application d'un niveau « 0 » sur l'entrée « NMI », effectue deux actions : la
30 premier est l'écriture d'un drapeau appelé « BLOQUE » dans la mémoire non

volatile programmable, et la seconde est une remise à zéro du microprocesseur. L'écriture du drapeau BLOQUE est irréversible ; l'état de ce drapeau ne peut plus être modifié ni par le microprocesseur, ni par un moyen extérieur. Un circuit d'initialisation, activé lors des mises sous tension du module de sécurité teste l'état du drapeau BLOQUE et bloque le microprocesseur dans un état de RESET permanent si le drapeau est écrit. Avantageusement, ce drapeau peut être réalisé par un fusible dont la fusion rend inutilisable le microprocesseur.

Si un fraudeur impose un niveau « 0 » à toutes les lignes du bus, y compris la ligne PARITE, le comparateur 12 ne détecte pas d'erreur. En effet, le nombre de lignes à « 0 » étant pair, le signal de parité doit être à « 0 » or, c'est justement à ce niveau que la ligne PARITE est forcée. Pour éviter cela et selon une variante de la seconde solution, on utilise le fait que les générateurs de parité ont généralement deux sorties, une pour générer une parité de type paire, l'autre pour générer une parité de type impaire.

La figure 5 montre le schéma électronique des modifications à apporter à la figure 4 pour intégrer dans le circuit monolithique des circuits générateurs de parité programmable. Les éléments communs à la figure 4 et à la figure 5 portent les mêmes références.

Les générateurs de parité 7a, 8a et 11a possèdent deux sorties : l'une notée Qp est la sortie du signal de parité paire, et l'autre Qi est la sortie du signal de parité impaire. Lorsque le nombre de lignes d'entrée à « 1 » est pair alors Qp est à « 1 » et Qi est à « 0 », lorsque le nombre de lignes d'entrée à « 1 » est impair alors Qi est à « 1 » et Qp est à « 0 ». Les montages 20 et 21 constitués chacun de deux portes ET, d'une porte OU et d'une porte inverseuse constituent des multiplexeurs. Les deux entrées des multiplexeurs sont respectivement connectées aux deux sorties Qi et Qp des générateurs de parité. Une ligne de contrôle notée 23 sélectionne l'entrée. Si la ligne 23 est à

« 0 », les sorties des portes 20c et 21c sont à « 0 » et, à l'aide des portes inverseuses 20b et 20c, les sorties des portes 20a et 21a ainsi que celles des portes 20d et 21d, reproduisent les niveaux logiques des sorties Qi des générateurs de parité 7a, 11a. Dans ce cas, les sorties Qi sont sélectionnées et les données présentes sur les sorties Qi sont envoyées au comparateur 12. Ce sont les signaux de parité impaire qui sont comparés. Si la ligne 23 est à « 1 », à l'aide des portes inverseuses 20b et 20c, les sorties des portes 20a et 21a sont à « 0 », les sorties des portes 20c et 21c, ainsi que celles des portes 20d et 21d, reproduisent les niveaux logiques des sorties Qp des générateurs de parité 7a et 11a. Dans ce dernier cas, les sorties Qp sont sélectionnées et les données présentes sur les sorties Qp sont envoyées au comparateur 12). Ce sont les signaux de parité paire qui sont comparés.

Les signaux de la ligne 23 sont envoyés par un générateur de signaux aléatoires 22). Ce générateur est un circuit électronique qui reçoit sur une ligne d'entrée l'horloge du microprocesseur et qui délivre un signal « 0 » et « 1 » de façon aléatoire dans le temps. De manière simplifiée, le générateur de signaux aléatoires peut être un compteur dont la sortie change d'état à chaque intervalle de temps déterminé. Le générateur de signaux aléatoires 22, le multiplexeur 21 et le comparateur 12 sont situés le plus près possible du microprocesseur, et de préférence intégrés à celui-ci. Ainsi, ils sont peu perturbés par un fraudeur émettant un rayonnement sur le bus. Des montages plus compliqués faisant appel à un oscillateur interne au générateur de signaux aléatoires peuvent être installés. Le but de ce générateur est d'émettre un signal logique sur une ligne dont l'état change assez souvent, de l'ordre de 100 à 10000 fois par seconde. Il est important de synchroniser les changements d'état de la ligne avec l'horloge du microprocesseur, ceci pour éviter de prendre en compte la ligne « NMI » au moment précis où le signal de la ligne 23 change, sinon il pourrait se produire des interférences dues aux différences de propagation des signaux.

En utilisation, Le fraudeur soumet le le bus de données à un rayonnement qui force les bits de données et de parité à « 1 » ou à « 0 ». Selon la valeur binaire, la valeur de parité calculée à partir des bits de données forcées par le rayonnement peut être égale à la valeur forcée par le rayonnement donc, étant égale, on ne détecte pas d'erreur. En changeant
 5 souvent le type de parité grace au générateur de signaux aléatoires, on détecte à coup sûr la présence du rayonnement.

Lorsque le générateur de signaux aléatoires 22 émet un niveau « 0 », les signaux émis par les sorties Qi des générateurs de parité 7a et 11a sont
 10 sélectionnés et donc, on compare les signaux de parité de type impaire. Lorsque le générateur de signaux aléatoires 22 émet un niveau « 1 », les signaux émis par les sorties Qp des générateurs de parité 7a et 11a sont sélectionnés et donc, les signaux de parité de type paire sont comparés.

Distinguons tout d'abord le cas où le bus est soumis à un rayonnement
 15 forçant les lignes du bus de données et de parité 2a à « 1 ». Lorsque le générateur de signaux aléatoires 22 envoie un signal « 0 », la sortie Qi du générateur de parité 11 est sélectionné, son niveau égal à « 0 » est différent de celui de la ligne de parité 2a qui est forcée à « 1 ». Le comparateur détecte donc bien ce type de rayonnement en déclenchant une interruption. Passons
 20 au deuxième cas où le bus est soumis à un rayonnement forçant les lignes du bus de données et de parité 2a à « 0 ». Lorsque le générateur de signaux aléatoires 2a2 envoie un signal « 1 », la sortie Qp du générateur de parité 11 est sélectionné, son niveau égal à « 1 » est différent de celui de la ligne de parité 2a qui est forcée à « 0 ». Dans ce cas également, le comparateur
 25 détecte le rayonnement et le signale au microprocesseur par une interruption.

Par cette variante, on ajoute un paramètre qui rend plus imprévisible encore le comportement du circuit pour un fraudeur, car ce paramètre augm nte la difficulté de contrôler l'état des lignes de bus par l'extérieur sans
 30 que le circuit le détecte.

Une troisième solution pour détecter une altération de l'exécution d'un programme est d'implémenter des routines de modification de drapeaux de place en place au sein d'un programme à protéger, et de vérifier avant
5 d'entreprendre une opération sur des données sensibles que tous les drapeaux ont été modifiés.

La figure 6 montre une partie de la mémoire contenant un programme implémenté à l'adresse 0800 hexadécimal. Cette mémoire peut être de la ROM
10 ou de l'EEPROM, mais tout autre type de mémoire non volatile capable d'exécuter des instructions convient. Tous les drapeaux sont représentés par des bits et regroupés en mémoire dans un indicateur. Dans l'exemple décrit, cet indicateur est un octet de la mémoire RAM appelé DRAPEAU. Un certain nombre de bits composant cet octet sont utilisés pour marquer le passage à
15 certaines étapes du programme qui mènent à une opération sur des données sensibles.

A l'adresse 800 (étape 1), l'octet DRAPEAU est mis à jour à la valeur binaire « 0000 0001 », le premier bit à « 1 » indiquant que l'étape 1 a été exécutée. A l'adresse 880H (étape 2), l'octet DRAPEAU est lu et modifié par
20 l'exécution d'un OU logique (instruction ORA en MOTOROLA 6805) entre son contenu actuel et la valeur binaire « 0000 0010 » : le résultat de l'opération OU est écrit dans l'octet DRAPEAU. A l'adresse 8A0H (étape 3), l'octet DRAPEAU est lu et modifié par l'exécution d'un OU logique entre son contenu et la valeur binaire « 0000 0100 » : le résultat de l'opération OU est écrit dans l'octet
25 DRAPEAU. Enfin, à l'adresse 900H (étape 4), la partie sécurisée du programme se termine : l'octet DRAPEAU est lu et contrôlé : si sa valeur est différente de la valeur binaire « 0000 0111 », le programme saute vers une routine de gestion de la fraude.

En utilisation, le programme à sécuriser commence à l'adresse 0800H. La première étape (étape 1) consiste à mettre à jour l'octet DRAPEAU en mettant à « 1 » le premier bit de l'octet DRAPEAU. Puis le programme continue en séquence jusqu'à une seconde étape dite de modification (étape 2) dans laquelle on positionne le second bit de l'octet DRAPEAU indiquant ainsi que l'étape 2 a été exécutée. Par voie de conséquence, on peut supposer que toutes les instructions du programme entre l'étape 1 et l'étape 2 ont été exécutées. Puis le programme continue en séquence jusqu'à l'étape de modification 3 où là, le troisième bit de DRAPEAU est mis à « 1 ». Enfin, le programme à sécuriser se termine par une routine de test de l'octet DRAPEAU (étape 4) : elle consiste à vérifier l'exécution des étapes 1,2 et 3. Si la valeur est différente de « 0000 0111 », un déroutement du programme est intervenu, ce qui révèle un fonctionnement anormal résultant très probablement d'une tentative de fraude. Dans ce cas, le programme interrompt son fonctionnement normal pour sauter vers la routine de gestion de la fraude. Un fraudeur ne connaissant pas les emplacements du programme où sont implémentées les routines de modification de l'octet DRAPEAU, il ne sait pas à quel moment elles s'exécutent et donc, en perturbant les valeurs du bus de données, il y a une forte probabilité pour qu'il réussisse à supprimer l'exécution d'au moins une des étapes 1,2 ou 3, et donc l'octet DRAPEAU n'aura pas la valeur finale attendue.

De façon simplifiée, la routine de gestion de la fraude peut consister en une remise à zéro du microprocesseur (RESET). Une amélioration consiste à utiliser pour l'étape 4 une instruction microprogrammée de telle sorte qu'un fraudeur ne pourrait perturber son déroulement en empêchant l'exécution de certaines instructions et en autorisant d'autres. Cette instruction a la structure suivante « Code opération, Adresse à lire, Valeur à comparer » : elle exécute séquentiellement les fonctions suivantes :

30

lecture de l'octet « Adresse »

- ② comparaison de la valeur de l'octet lue avec « Valeur »
- ③ si égal alors saut à l'instruction suivante
- sinon ④ écriture du drapeau BLOQUE en mémoire non volatile
- ⑤ RESET du microprocesseur

5

Bien évidemment, en augmentant le nombre de drapeaux et donc le nombre d'étapes de mise à jour de l'indicateur, on augmente les moments de détection des perturbations extérieures. L'indicateur DRAPEAU doit alors être représenté par plusieurs octets. Mais, les étapes de mise à jour de l'indicateur occupent de la place mémoire inutile pour le programme d'application, tant au

10 niveau du programme qu'au niveau de la mémoire RAM. Il faut donc optimiser le nombre d'étapes. Par exemple, pour un programme à sécuriser de 1000 octets, un bon compromis serait d'installer 32 étapes de modification de l'indicateur. Ces 32 étapes et la routine de test final occupent 162 octets de

15 mémoire de programme et 4 octets en RAM. Selon la complexité du programme, qui peut comporter des sauts et ne pas effectuer toutes les étapes de modification, le test de l'octet DRAPEAU peut ne prendre en compte qu'un nombre limité de bits. Si le programme à sécuriser se termine à des endroits différents, on peut installer dans le programme plusieurs routines de test qui

20 prennent en compte des valeurs différentes de l'octet DRAPEAU.

Cette solution comporte l'avantage d'être facilement utilisable sur un composant classique car elle n'implique pas de modifier la partie matérielle du composant.

25

Un perfectionnement de l'invention consiste à implémenter dans le programme des instructions d'effacement de l'octet DRAPEAU à des emplacements qui ne sont normalement jamais atteints lors de l'exécution du programme. Ainsi, une perturbation frauduleuse de l'exécution du programme peut provoquer l'exécution d'une de ces instructions qui, en mettant à 00 l'octet

30 DRAPEAU, entraîne l'exécution de la routine de gestion de la fraude.

La figure 7 montre l'aspect de la mémoire de programme selon le perfectionnement. A l'adresse 0890H (étape 2bis), le programme exécute une instruction de saut inconditionnel. L'instruction à l'adresse suivante n'est donc jamais exécutée, sauf si une autre instruction de saut la spécifie comme destination. Le programmeur écrit une instruction d'effacement de l'octet DRAPEAU juste après l'instruction de saut inconditionnel, et prend bien soin de ne jamais la spécifier comme destination dans son programme. Si un fraudeur perturbe le bus de données, il y a un certain niveau de probabilité pour que l'instruction de saut inconditionnel ne soit pas exécutée et que cette instruction d'effacement le soit. Cette instruction met à « 0 » tous les bits de l'octet DRAPEAU. Lors du test final à l'étape 5, les bits 1 et 2 de DRAPEAU sont à « 0 » et donc la valeur lue n'est pas celle attendue. Le programme interrompt donc son fonctionnement normal pour sauter vers la routine de gestion de la fraude.

Le programme illustré par la figure 7 montre deux étapes d'effacement de l'octet DRAPEAU (étape 2 bis et 3 bis). L'instruction d'effacement n'occupe que deux octets en mémoire de programme, contre quatre octets pour les étapes de mise à jour de l'indicateur, ce qui, à performance égale, fait gagner de la place. Un programme optimal utilise assez peu d'instructions de saut inconditionnel. Il est donc possible de mettre systématiquement une instruction d'effacement après un saut inconditionnel.

REVENDEICATIONS

1. Procédé de sécurisation du traitement d'une information sensible dans
5 un module de sécurité de structure monolithique, le module comportant des
moyens de traitement de l'information (31) et des moyens de mémorisation
(32,33) d'informations susceptibles d'être traitées par lesdits moyens de
traitement, caractérisé en ce qu'il comprend les étapes suivantes :
- sélectionner une information sensible dans les moyens de
10 mémorisation ;
 - déterminer une condition particulière d'intégrité de la dite information ;
 - lire l'information et la transmettre aux moyens de traitement ;
 - contrôler lors du traitement de l'information que la condition particulière
est satisfaite ;
 - 15 - bloquer le traitement de l'information au cas où la condition particulière
n'est pas satisfaite.
2. Procédé selon la revendication 1, dans lequel l'information est un code
opération lu dans les moyens de mémorisation (32,33), l'ensemble des codes
20 opérations étant contenu dans une table ayant un contenu déterminé lors de la
fabrication du module de sécurité, et la condition particulière d'intégrité est le
fait que la valeur de l'information est égale à l'une de plusieurs valeurs fixes.
3. Procédé selon la revendication 2, dans lequel le code opération à traiter
25 est codé sous forme de bits de données et lesdits bits n'ont pas tous la même
valeur binaire.
4. Procédé selon la revendication 1, dans lequel la condition particulière
d'intégrité consiste à contrôler une donnée d'intégrité calculée en utilisant
30 l'information lue dans les moyens de mémorisation (32,33), la donnée
d'intégrité étant calculée lors de la lecture de l'information et étant transmise
-

aux moyens de traitement, les moyens de traitement calculant une autre donnée d'intégrité à partir des informations reçues et contrôlant l'égalité entre les deux données d'intégrité.

5 5. Procédé selon la revendication 4, dans lequel les données d'intégrité sont calculées à partir d'au moins une donnée de calcul dont la valeur varie en fonction du temps.

10 6. Procédé selon la revendication 4, dans lequel les données d'intégrité sont calculées à partir d'au moins une donnée de calcul dont la valeur varie de façon aléatoire.

7. Procédé selon la revendication 1, dans lequel le blocage du traitement de l'information est réalisé par une instruction microprogrammée.

15

8. Procédé selon la revendication 7, dans lequel l'instruction microprogrammée réalise les étapes suivantes :

- écrire une donnée de blocage dans un emplacement non volatile des moyens de mémorisation (32,33) ;

20

- bloquer le traitement de l'information.

9. Procédé selon la revendications 8, dans lequel, à la mise sous tension du module, un emplacement non volatile des moyens de mémorisation (32,33) est lu par les moyens de traitement (31), et le module est bloqué si une valeur
25 lue à cet emplacement n'est pas conforme.

10. Module de sécurité constitué d'un circuit électronique de structure monolithique et comportant des moyens de traitement de l'information (31) et des moyens de mémorisation (32,33), les moyens de traitement sélectionnant
30 des informations extraites des moyens de mémorisation afin d les traiter ;

caractérisé en ce que les moyens de traitement comportent des moyens de contrôle d'une condition particulière d'intégrité d'une information sensible, et des moyens de blocage du traitement de l'information, lesdits moyens de blocage étant activés lorsque les moyens de contrôle ont détecté que la
5 condition particulière n'est pas satisfaite.

11. Module de sécurité selon la revendication 10, dans lequel les moyens de traitement (31) exécutent des instructions correspondant à des codes opérations extraits d'une table, caractérisé en ce que la table comprend une
10 valeur d'instruction interdite.

12. Module de sécurité selon la revendication 11, dans lequel le code opération à traiter est codé sous forme de bits de données, le module de sécurité comprenant un moyen de lecture des valeurs de tous les bits et un
15 moyen de blocage activé lorsque les valeurs des bits sont toutes identiques.

13. Module de sécurité selon la revendication 10, dans lequel les moyens de traitement (31) exécutent des instructions correspondant à des codes opérations extraits d'une table, le module de sécurité comportant un moyen de
20 lecture d'un code opération et un moyen de blocage activé lors de la lecture d'un code opération interdit.

14. Module de sécurité selon la revendication 13, dans lequel le moyen de blocage comprend un moyen d'écriture irréversible d'un indicateur dans les
25 moyens de mémorisation (32,33), et un moyen de lecture dudit indicateur lors de la mise sous tension ultérieure du module.

15. Module de sécurité selon la revendication 10, comportant des générateurs de parité (7,8) coopérant avec les moyens de mémorisation, des
30 gén'rateurs de parité (11) coopérant avec le moyen de traitement et un

comparateur relié à chacun des générateurs de parité et apte à provoquer une interruption au sein des moyens de traitement.

- 5 16. Module de sécurité selon la revendication 15, dans lequel les générateurs de parité (7,8) ont un fonctionnement qui varie en fonction du temps.
17. Module de sécurité selon la revendication 15, dans lequel les générateurs de parité (7,8) ont un fonctionnement qui varie aléatoirement.
- 10 18. Module de sécurité selon la revendication 14, caractérisé en ce que l'écriture irréversible de l'indicateur dans les moyens de mémorisation (32,33) est réalisée en exécutant une instruction microprogrammée.
- 15 19. Module de sécurité selon la revendication 10, caractérisé en ce que le module de sécurité est une carte à microcircuit.
-

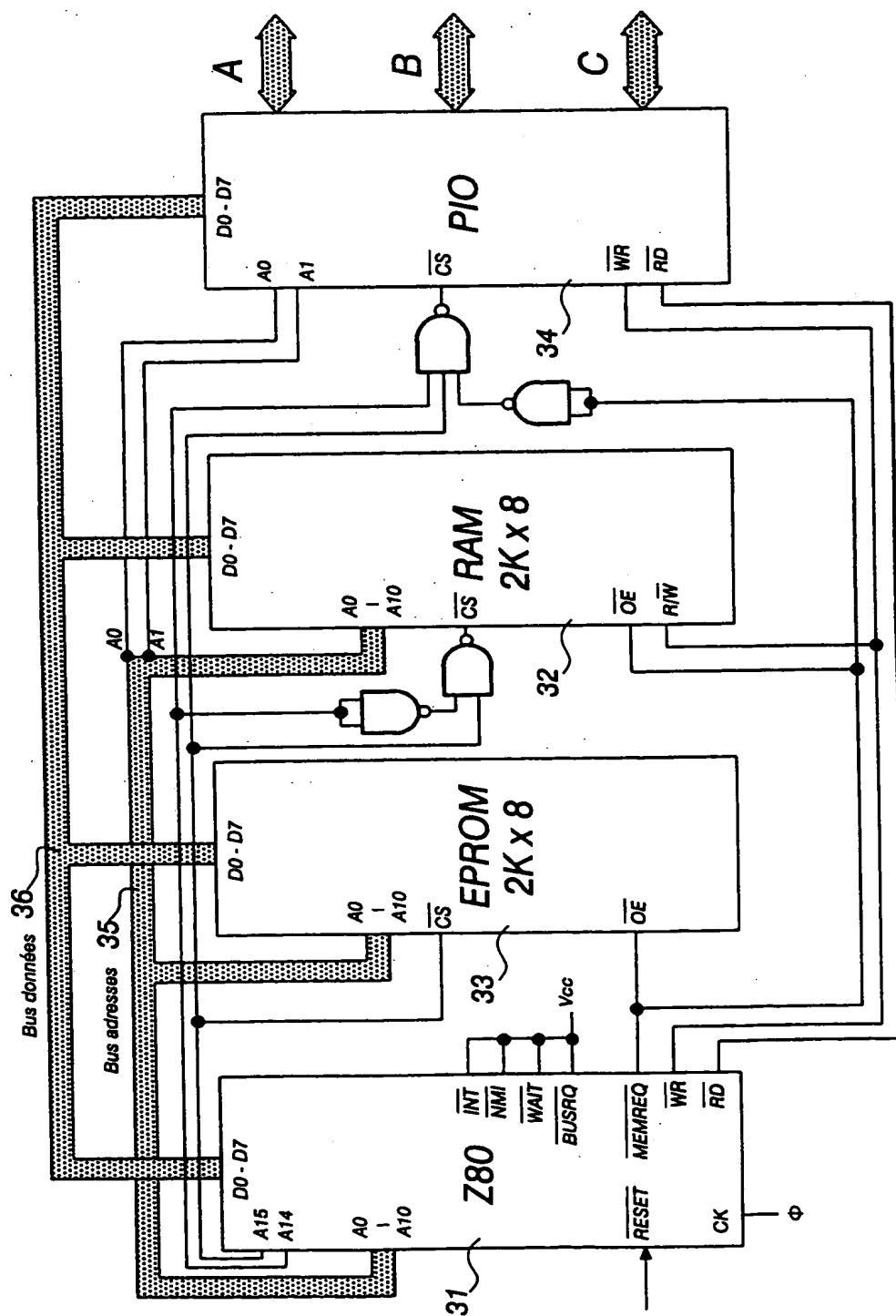


Fig.1

\Poids forts	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
\Poids faibles	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0 0000	BRSET0	BSET0	BRA	NEG	NEG	NEG	NEG	NEG	RTI		SUB	SUB	SUB	SUB	SUB	SUB
1 0001	BRSET1	BSET1	BRN						RTS		CMP	CMP	CMP	CMP	CMP	CMP
2 0010	BRSET2	BSET2	BHI								SBC	SBC	SBC	SBC	SBC	SBC
3 0011	BRSET3	BSET3	BLS	COM	COM	COM	COM	COM	SWI		CPX	CPX	CPX	CPX	CPX	CPX
4 0101	BRSET4	BSET4	BCC	LSR	LSR	LSR	LSR	LSR			AND	AND	AND	AND	AND	AND
5 0101	BRSET5	BSET5	BCS								BIT	BIT	BIT	BIT	BIT	BIT
6 0110	BRSET6	BSET6	BNE	ROR	ROR	ROR	ROR	ROR			LDA	LDA	LDA	LDA	LDA	LDA
7 0111	BRSET7	BSET7	BEQ	ASR	ASR	ASR	ASR	ASR		TAX		STA	STA	STA	STA	STA
8 1000	BRCLR0	BCLR0	BHCC	LSL	LSL	LSL	LSL	LSL		CLC	EOR	EOR	EOR	EOR	EOR	EOR
9 1001	BRCLR1	BCLR1	BHCS	ROL	ROL	ROL	ROL	ROL		SEC	ADC	ADC	ADC	ADC	ADC	ADC
A 1010	BRCLR2	BCLR2	BPL	DEC	DEC	DEC	DEC	DEC		CLI	ORA	ORA	ORA	ORA	ORA	ORA
B 1011	BRCLR3	BCLR3	BMI							SEI	ADD	ADD	ADD	ADD	ADD	ADD
C 1100	BRCLR4	BCLR4	BMC	INC	INC	INC	INC	INC		RSP		JMP	JMP	JMP	JMP	JMP
D 1101	BRCLR5	BCLR5	BMS	TST	TST	TST	TST	TST		NOP	BSR	JSR	JSR	JSR	JSR	JSR
E 1110	BRCLR6	BCLR6	BIL								LDX	LDX	LDX	LDX	LDX	LDX
F 1111	BRCLR7	BCLR7	BIH	CLR	CLR	CLR	CLR	CLR	WAIT	TXA		STX	STX	STX	STX	STX

Fig.2

\Poids forts \Poids faibles	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 0000	FRD	RTI	BSET0	BSET0	BRA	NEG	NEG	NEG	NEG	1001	SUB	SUB	SUB	SUB	SUB	SUB
1 0001	FRD	RTS	BSET1	BSET1	BRN	FRD	FRD	FRD	FRD	FRD	CMP	CMP	CMP	CMP	CMP	CMP
2 0010	FRD	FRD	BSET2	BSET2	BHI	FRD	FRD	FRD	FRD	FRD	SBC	SBC	SBC	SBC	SBC	SBC
3 0011	FRD	SWI	BSET3	BSET3	BLS	COM	COM	COM	COM	COM	CPX	CPX	CPX	CPX	CPX	CPX
4 0101	FRD	FRD	BSET4	BSET4	BCC	LSR	LSR	LSR	LSR	LSR	AND	AND	AND	AND	AND	AND
5 0101	FRD	FRD	BSET5	BSET5	BCS	FRD	FRD	FRD	FRD	FRD	BIT	BIT	BIT	BIT	BIT	BIT
6 0110	FRD	FRD	BSET6	BSET6	BNE	ROR	ROR	ROR	ROR	ROR	LDA	LDA	LDA	LDA	LDA	LDA
7 0111	TAX	FRD	BSET7	BSET7	BEQ	ASR	ASR	ASR	ASR	ASR	STA	STA	STA	STA	STA	STA
8 1000	CLC	FRD	BCLR0	BCLR0	BHCC	LSL	LSL	LSL	LSL	LSL	EOR	EOR	EOR	EOR	EOR	EOR
9 1001	SEC	FRD	BCLR1	BCLR1	BHCS	ROL	ROL	ROL	ROL	ROL	ADC	ADC	ADC	ADC	ADC	ADC
A 1010	CLI	FRD	BCLR2	BCLR2	BPL	DEC	DEC	DEC	DEC	DEC	ORA	ORA	ORA	ORA	ORA	ORA
B 1011	SEI	FRD	BCLR3	BCLR3	BMI	FRD	FRD	FRD	FRD	FRD	ADD	ADD	ADD	ADD	ADD	ADD
C 1100	RSP	FRD	BCLR4	BCLR4	BMC	INC	INC	INC	INC	INC	JMP	JMP	JMP	JMP	JMP	JMP
D 1101	NOP	FRD	BCLR5	BCLR5	BMS	TST	TST	TST	TST	TST	JSR	JSR	JSR	JSR	JSR	JSR
E 1110	FRD	FRD	BCLR6	BCLR6	BIL	FRD	FRD	FRD	FRD	FRD	LDX	LDX	LDX	LDX	LDX	LDX
F 1111	TXA	WAIT	BCLR7	BCLR7	BIH	CLR	CLR	CLR	CLR	CLR	STX	STX	STX	STX	STX	STX

Fig.3

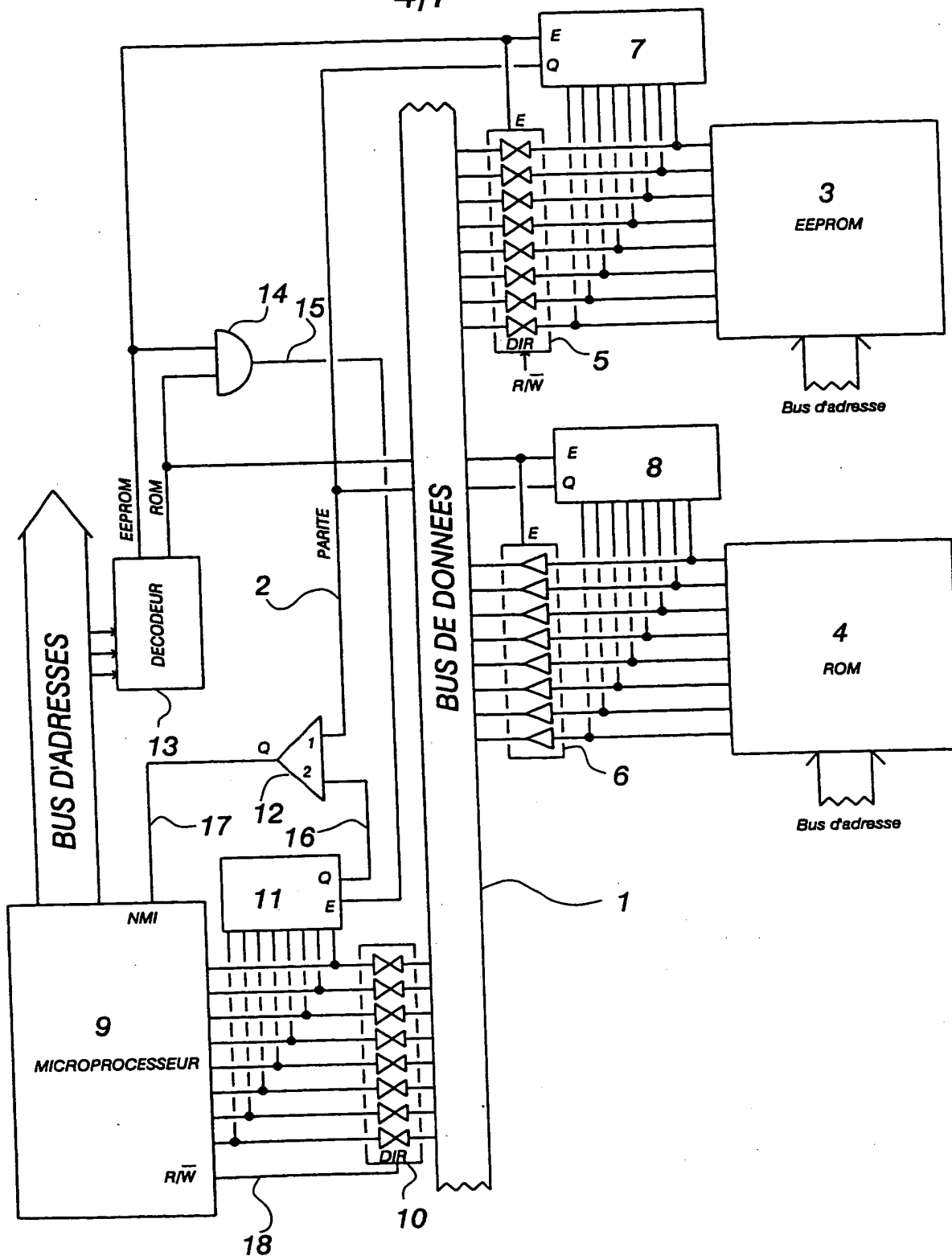


Fig. 4

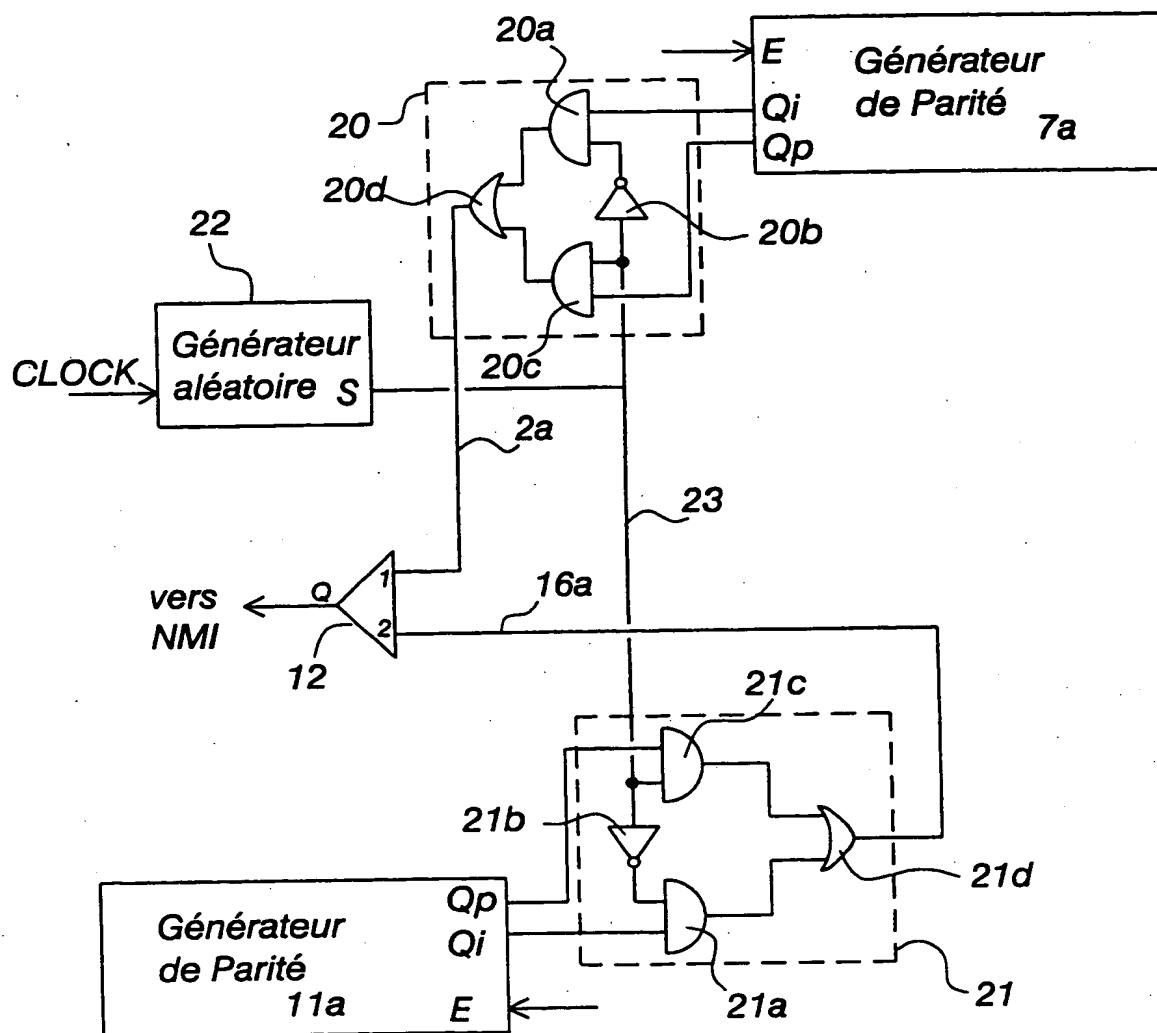


Fig.5

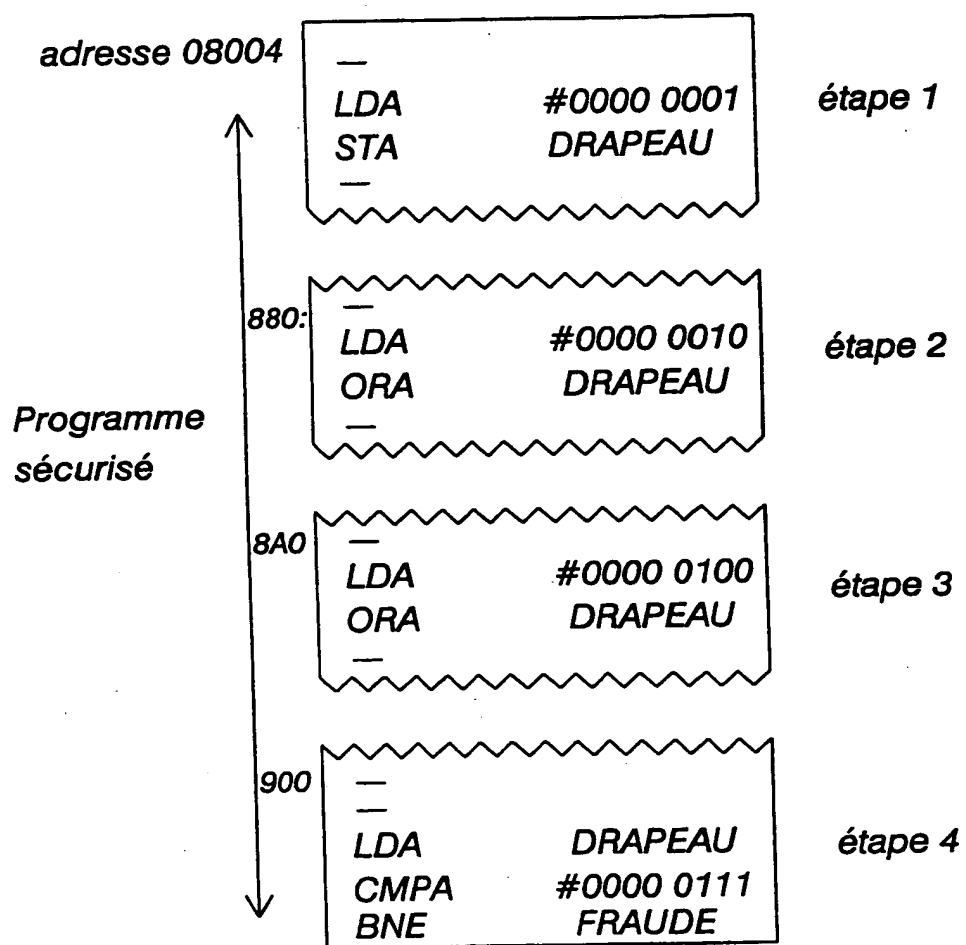


Fig.6

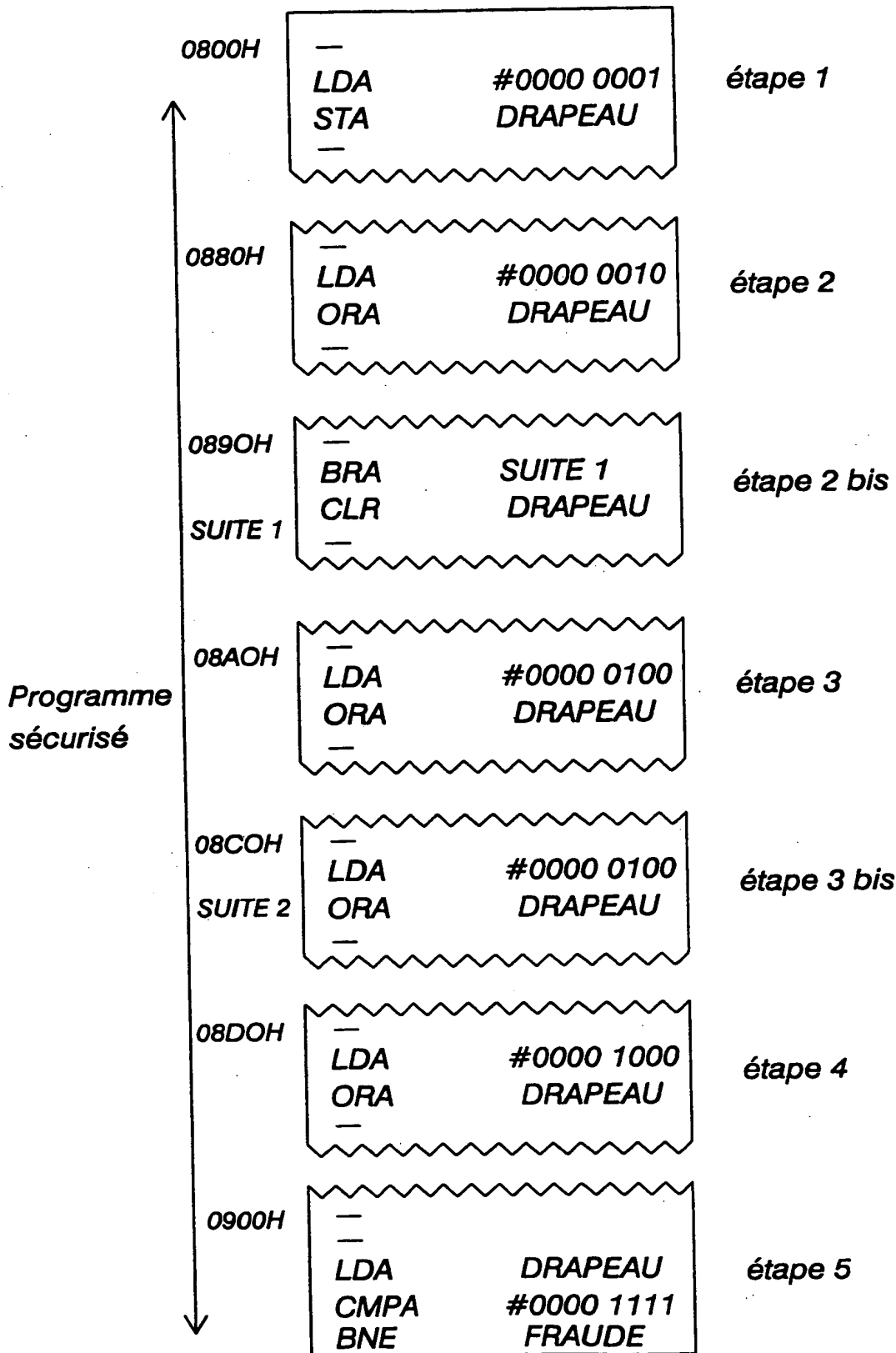


Fig.7

This Page Blank (uspto)
